



A Generic High-performance GPU-based Library for PDE solvers

Glimberg, Stefan Lemvig; Engsig-Karup, Allan Peter

Publication date:
2012

Document Version
Publisher's PDF, also known as Version of record

[Link back to DTU Orbit](#)

Citation (APA):
Glimberg, S. L., & Engsig-Karup, A. P. (2012). *A Generic High-performance GPU-based Library for PDE solvers*. Abstract from International Workshop PDE Software Frameworks, Münster, Germany.

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

A Generic High-performance GPU-based Library for PDE solvers

Stefan L. Glimberg and Allan P. Engsig-Karup

Technical University of Denmark

Recent development of massively parallel processors such as graphical processing units (GPUs), has already proven to be very effective for a vast amount of scientific applications. One major benefit of the GPU, is that it is already a standard device in most affordable desktop computers. Thus, the privilege of high-performance parallel computing is now in principle accessible for many scientific users, no matter their economic resources. Though being highly effective units, GPUs and parallel architectures in general, pose challenges for software developers to utilize their efficiency. Sequential legacy codes are not always easily parallelized and the time spent on conversion might not pay off in the end. We present a highly generic C++ library for fast assembling of partial differential equation (PDE) solvers, aiming at utilizing the computational resources of GPUs. The library requires a minimum of GPU computing knowledge, while still offering the possibility to customize user-specific solvers at kernel level if desired.

Spatial differential operators are based on matrix free flexible order finite difference approximations. These matrix free operators minimize both memory consumption and main memory access, two important features for efficient GPU utilization and for enabling solution of large problems. In order to solve the large linear systems of equations, arising from the discretization of PDEs, the library includes a set of common iterative solvers. All iterative solvers are based on template arguments, such that vector and matrix classes, along with their underlying implementations, can be freely interchanged or new schemes developed without much coding effort. The generic nature of the library, along with a predefined set of interface rules, allow us to set up the components for PDE solver through type binder definitions. We encourage this use of parameterized binding objects, as it allows the user to control the assembling of PDE solvers at a high abstraction level, without necessarily having to change internal code.

We will illustrate the assembling of a tool using our library for fast and scalable simulation of fully nonlinear free surface water waves over uneven depths[1, 2, 3]. The wave model is based on the potential flow formulation, with the computational bottleneck of solving a fully three dimensional Laplace problem efficiently. A robust h - or p -multigrid preconditioned defect correction method is applied to keep storage low and algorithmic efficiency high. Performance analysis of the implemented wave model shows that performance is comparable to a dedicated (non-library version) reference GPU-based solver. Work in progress also address the problem of simulating water waves at very large scales. Therefore we added an MPI layer to support domain decomposition preconditioning using multiple GPUs. The wave tool is to be used for efficient analysis of both coastal engineering problems and interactive real-time computing of ship-wave problems. Such applications will benefit well from high-performance software. We will report our recent progress on the development of the new tool for coastal engineering.

References

- [1] A. P. Engsig-Karup, H. B. Bingham, O. Lindberg: An efficient flexible-order model for 3D nonlinear water waves. *Journal of Comp. Phys.* 228 (2012) 2100-2118.
- [2] A. P. Engsig-Karup, S. L. Glimberg, M. G. Madsen: A massively parallel GPU-accelerated model for analysis of fully nonlinear free surface waves. *Int. J. Num. Meth. Fluids.* (2011).
- [3] S. L. Glimberg, A. P. Engsig-Karup, M. G. Madsen: A Fast GPU-accelerated Mixed-precision Strategy for Fully Nonlinear Water Wave Computations. *Proceedings ENUMATH.* (2012).